SLOW MOTION PROCESSING OF DIGITAL VIDEO DATA

Lu Zhen

Yushan Huang

Donghui Wu

Lingxiang Zhou

FIELD OF INVENTION

[0001] This invention relates to a method for generating a slow motion effect in a video.

DESCRIPTION OF RELATED ART

[0002] In order to enhance the visual effect of a motion scene, slow motion processing can construct and insert new intermediate frames between each pair of original frames. During playback, the processed video produces a "slow motion" effect to the viewers.

[0003] It is well known that simple frame reconstruction techniques such as frame repetition or linear interpolation introduce annoying artifacts. Frame repetition generates jerky object motions because object movements are simply not considered and thus not accounted for. Linear interpolation by temporal filtering exhibits blurring in moving areas because object motions are not considered and pixel values in different object regions used in the interpolation result in the blurring in object region boundaries. Object motion must be compensated in order to remove these artifacts.

[0004] Motion compensated temporal interpolation (MCTI) techniques can be used in slow motion processing of digital video data to construct new intermediate frames with considerable less artifacts. Motion estimation and compensation is a powerful means of exploiting the temporal redundancy contained in video sequences. This means is widely used in most video applications, such as video coding, de-interlacing, de-noising, de-bluring, etc. In motion compensated temporal interpolation (MCTI), the principal idea is to reconstruct all pixels at a certain time instant of their motion trajectory. An accurate interpolation requires the estimation of "true" (i.e., actual) motion vectors.

[0005] Many motion estimation techniques have been investigated. Block matching method is the most popular one, especially in video coding applications. The main advantages are its simplicity, low computational complexity, and low overhead. However, block matching produces inaccurate motion field that are piecewise constant and are not usually representative of the true motion. Video coders employ this crude motion estimation method in order to keep the bit-overhead low. The interpolated frames usually contain severe blocking artifacts and are visually inadequate, thereby necessitating the encoding and transmission of residuals for the B-frame in MPEG standard. However, in slow motion processing, motion estimates that are accurate and close to the "true" motion are expected. This is because prediction residuals are not available in this case.

[0006] Thus, what is needed is a method for producing a slow motion effect that addresses the disadvantages described above.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] Fig. 1 illustrates a method for generating slow motion effect in one embodiment of the invention.

[0008] Fig. 2 illustrates an image pyramid for generating slow motion effect in one embodiment of the invention.

[0009] Fig. 3 illustrates a pyramidal method for estimating motion in one embodiment of the invention.

[0010] Fig. 4 illustrates an iterated registration method for estimating motion in one embodiment of the invention.

[0011] Fig. 5 illustrates a method for generating an intermediate frame from a motion field between two consecutive frames in one embodiment of the invention.

[0012] Fig. 6 is a flowchart of a method for generating a slow motion effect in one embodiment of the invention.

[0013] Use of the same reference numbers in different figures indicates similar or identical elements.

SUMMARY

[0014] In one embodiment of the invention, a method includes (1) generating a first image pyramid of a first image, (2) generating a second image pyramid of a second image, (3) warping a first level image of the first image pyramid with a motion field, (4) determining a residual motion field from the warped first level image of the first image pyramid and a corresponding first level image of the second image pyramid, and (5) if the residual motion field is not less than a threshold, adding the residual motion field to the motion field and repeating steps (3) and (4).

DETAILED DESCRIPTION

[0015] In accordance with the invention, a robust and accurate motion compensated temporal interpolation (MCTI) technique is applied in slow motion processing of digital video data to construct new intermediate frames with considerable less artifacts. As shown in Fig. 1, the slow motion processing 10 is divided into two stages: motion estimation and motion compensation. An accurate and dense motion field can be determined from each pair of consecutive frames in the original sequence. With the motion field, pixels in the original frame can be moved to appropriate locations along the motion trajectories to form a new intermediate frame. The new slow motion processed video is then formed by inserting the new intermediate frames between the original frames.

[0016] In one embodiment of the invention, the motion estimation algorithm disclosed by Horn and Schunck is used to determine a motion field between frames. B.K.P Horn, B.G. Schunck, "Determining Optical Flow," Massachusetts Institute of Technology Artificial Intelligence Memo No. 572, April 1980. As a gradient based motion estimation method, the Horn and Schunck (HS) algorithm does not properly handle large displacement due to a linear Taylor series approximation used in the algorithm. Two modifications to the basic HS algorithm are introduced in accordance with the invention. One modification is the use of multi-resolution measurements from an image pyramid. The other modification is the use of iterated registration in motion field computation at each level of the image pyramid.

Pyramidal Motion Estimation Algorithm

[0017] In one embodiment of the invention, a coarse-to-fine strategy is used in a pyramidal motion estimation algorithm. Two image pyramids of the two frames, between which the motion field is to be determined, are constructed by successive low-pass filtering and sub-sampling. In one embodiment, the coding algorithm disclosed by Burt and Adelson is used to construct Laplacian image pyramids of the two frames. Peter J. Burt and Edward H. Adelson, "The Laplacian Pyramid as a Compact Image Code," IEEE Transactions on Communications, Vol. Com-31, No. 4, April 1983. Low resolution motion can then be estimated reliably at the coarse level of the image pyramid. However, the loss of high frequency components makes it difficult to estimate high resolution motion.

[0018] A possible remedy consists in first passing the coarse motion field to the next finer level, and then using the coarse motion field as an initial guess for the motion field at the next finer level. Specifically, the coarse motion field is used to warp (to motion compensate) one of the two frames in the next finer level (e.g., by linearly interpolating the coarse motion field to provide a motion vector for each pixel in the next level). At the next finer level, the residual motion between the two frames is now smaller. Thus, the high frequency components can now be used to more reliably estimate fine corrections (motion field refinements) to the coarse motion field. The corrected motion field can then be passed from level to level until the finest level.

[0019] Fig. 2 illustrates an image pyramid 30 having i_{max} (e.g., 3) number of levels in one embodiment. The motion estimation begins at the highest level $L^{i_{max}}$, where a coarse motion field $d^{i_{max}}$ is obtained using an iterative motion estimator. The iterative motion estimation algorithm is detailed in the next section. The coarse motion field $d^{i_{max}}$ is then propagated to next finer level $L^{i_{max}-1}$ in as an initial guess for the motion field in the iterative motion estimation at level $L^{i_{max}-1}$. As shown in Fig. 3, at each pyramid level L^{i} of frames I_{t-1} and I_t, the motion field d^{i+1} is propagated from the coarser level L^{i+1} and used as an initial guess for the motion field. Given that initial guess, the refined motion field is computed by the iterative motion estimation, and the result is propagated to the next finer level L^{i-1}, and so on to level L^{0}, which represents the original frame. The final result d^{0} is the desired motion field between frames I_{t-1} and I_t.

Iterative Motion Estimation Algorithm

[0020] When the motion between frames I_{t-1} and I_t is very large, the pyramidal motion estimator will require many levels in the image pyramid. This can lead to over-smoothing at the coarse levels that cannot be corrected at the finer levels, since the HS algorithm can only estimate small corrections. In this situation, an iterated registration method disclosed by Lucas and Kanade is added to the HS algorithm at each level of the image pyramid. B. Lucas, T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," In Proceedings of the 7th International Joint Conference on Artificial Intelligence, 1981. The coarse-to-fine strategy is used again here. The coarse motion field is used to warp one of the two frames, and the smaller residual motion between the two frames (one warped and the other unchanged) is computed using the HS algorithm, and added to the coarse motion field as a refinement. The warping and the computing the residual motion can be repeated to get a more refined motion field at each level of the image pyramid.

[0021] The difference to the coarse-to-fine strategy used in pyramidal motion estimation algorithm described in the last section is that the motion field is passed within the level, not from coarse to finer levels. As shown in Fig. 4, at level L^i, the coarse motion field d^{i+1} of level L^{i+1} is propagated and used as an initial guess $d^{i'}$ for the motion field. Frame I^i_{t-1} is then warped to I'^i_{t-1} by the initial guess $d^{i'}$. Using the HS algorithm, the residual motion *r* between warped frame I'^i_{t-1} and frame I^i_t is determined, and added to the initial guess $d^{i'}$ as a refinement. The refined motion field is then used as initial guess again. The procedures of warping frame, the HS motion estimation, the motion field refining are carried out recursively, until the norm of the residual motion field *r* is less than a predefined threshold R_{thre}, or the iterative number *n* is more than a predefined threshold N_{thre}. The final result of the motion field at level L^i is propagated to next finer level L^{i-1} as the initial guess of that level according to the pyramidal motion estimation algorithm described in last section.

[0022] The above described motion estimation method combines the iterated registration method with the pyramidal motion estimation method. This method, hereafter referred as iterative pyramidal motion estimation (IPME), has two major advantages. Firstly, lesser number of levels in the image pyramid will be needed since larger motion at each level can now be track. Secondly, the coarse motion estimation errors propagated to the finer levels can be recovered. At the same time, IPME algorithm has faster convergence property than that of the HS algorithm, and it is more efficient than the HS algorithm.

Motion Compensation

[0023] After motion estimation between frames I_{t-1} and I_t , a dense and accurate motion field d, which is the final result of motion field d^0 at level L^0, is determined. With the motion vectors in motion field d, a matching pixel in frame I_t is found for each pixel in frame I_{t-1}. Then, along the motion trajectory, the matched pixels pair is moved to a proper pixel location on the intermediate frame I_{int} as shown in Fig. 5. In Fig. 5, λ is a parameter representing the location on the motion trajectory from frame I_{t-1} to frame I_t , where λ ranges from 0 (at a corresponding pixel location in frame I_{t-1}) to 1 (at a corresponding pixel location in frame I_t). Thus, a motion vector is assigned that pixel location on the frame I_{int}.

[0024] Most pixels in frame I_{int} can be assigned one motion vector. A few pixels in frame I_{int} will have multiple assignments. These can be handled by averaging. A few pixels in frame I_{int} may receive no assignment. For these pixels, the motion vectors of the neighboring pixels are fitted to an affine translation using least-squares methods. Then the motion vectors for these pixels are computed by the fitted affine translation.

[0025] After the assignment of the motion vectors, the value of each pixel in frame I_{int} can be computed from the matched pixels pair. The color value of each pixel in frame I_{int} is computed by linear interpolation of the matched pixel pair according to location parameter λ.

Exemplary flowchart

[0026] Fig. 6 illustrates a flowchart of a method 100 for implementing the motion estimation and motion compensation described above in one embodiment of the invention. Method 100 can be used to generate an intermediate frame I_{int} between frames I_{t-1} and I_t . When method 100 is performed to an entire video sequence, a slow motion effect is achieved when the video sequence is played back. Method 100 can be implemented with software on a computer or any equivalents thereof.

[0027] In step 102, the computer selects two sequential frames I_{t-1} and I_t from a video sequence.

[0028] In step 104, the computer generates image pyramids of frames I_{t-1} and I_t . In one embodiment, the computer generates Laplacian image pyramids as disclosed by Burt and Adelson.

[0029] In step 106, the computer selects images at the coarsest level ($L^{i_{max}}$) of the image pyramids for frames I_{t-1} and I_t .

[0030] In step 108, the computer estimates a motion field *d* between frames I_{t-1} and I_t from their top levels images. In one embodiment, the computer determines motion field *d* going from frame I_{t-1} to frame I_t . In one embodiment, the computer estimates the motion field *d* using the HS algorithm as disclosed by Horn and Schunck.

[0031] In step 110, the computer warps frame I_{t-1} at the current image level with motion field *d* to form a warped frame I_{t-1}.

[0032] In step 112, the computer estimates a motion field *r* (hereafter "residual motion field *r*") going from warped frame I_{t-1} to frame I_t at the current image level. In one embodiment, the computer estimates residual motion field *r* using the HS algorithm as disclosed by Horn and Schunck.

[0033] In step 114, the computer determines if the norm of residual motion field *r* (i.e., $||r||$) is less than a threshold R_{thre} or if an iterative number *n* of times through the loop consisting of steps 110, 112, 114, and 116 is greater than a threshold N_{thre}. If none of these conditions is true, then step 114 is followed by step 116. Otherwise step 114 is followed by step 118.

[0034] In step 116, the computer adds residual motion field *r* to motion field *d.* Step 116 is followed by step 110 and this loop repeats to further refine motion field *d.*

[0035] In step 118, the computer determines if the current iteration has processed the finest level (L_0) of the image pyramids. If not, then step 118 is followed by step 120. Otherwise step 118 is followed by step 122.

[0036] In step 120, the computer selects corresponding images at the next finer level of the image pyramids for frames I_{t-1} and I_t . Step 120 is followed by step 110 and method 100 repeats until all the levels of the image pyramids have been processed.

[0037] In step 122, the computer generates intermediate frame I_{int} from motion field *d.*

[0038] In step 124, the computer inserts intermediate frame I_{int} between frames I_{t-1} and I_t in the video sequence.

Conclusions

[0039] After the procedures of motion estimation and motion compensation for each pair of consecutive frames in the original video sequence, one or more new intermediate frames can be generated and inserted into the sequence. A new video sequence with increased temporal resolution is achieved. It will exhibit slow motion effect during playback at the same frame rate as the original video sequence.

[0040] On the other hand, if the processed video is played in the same time length as the original video sequence, the frame rate is up-converted and a "fast motion" effect is created. This invention can also be used in other applications of video data, like coding, de-interlacing, debluring, de-noising, etc.

[0041] Various other adaptations and combinations of features of the embodiments disclosed are within the scope of the invention. Numerous embodiments are encompassed by the following claims.